

Amendments to the Claims

The following listing of claims will replace all prior versions and/or listings of claims in the application.

Listing of Claims:

1. (Previously presented) A method of reducing a size of data difference representations, the method comprising:

identifying an original version of an input data stream in an original form;

dividing the original form of the original version of the input data stream into separate original version output data streams through the use of a pre-processor;

identifying an updated version of the input data stream in the original form;

dividing the original form of the updated version of the input data stream into separate updated version output data streams through the use of the pre-processor; and

differencing at least two of the separate updated version output data streams with a corresponding separate original version output data stream to produce data difference representations.

2. (Original): The method of claim 1, wherein the data difference representations are smaller than a data difference representation created by differencing the original form of the updated version of the input data stream with the original form of the original input data stream.

3. (Previously presented): The method of claim 1, further comprising:

reconstructing the separate updated version output data streams from the data difference representations and the original version output data streams; and

combining the separate updated version output data streams into the original form of the updated version of the input data stream through the use of a post-processor.

4. (Original): The method of claim 1 wherein the original form of the original version of the input data stream is empty.
5. (Original): The method of claim 1 wherein the pre-processor comprises decompression algorithms.
6. (Previously presented): The method of claim 1 wherein the dividing separates volatile components of the input data stream from less volatile components of the input data stream.
7. (Original): The method of claim 6 wherein the input data stream is executable code.
8. (Original): The method of claim 7 wherein the volatile components comprise branch targets.
9. (Original): The method of claim 7 wherein the volatile components comprise data addresses.
10. (Original): The method of claim 7 wherein the less volatile components comprise instruction code.
11. (Original): The method of claim 7 wherein the less volatile components comprise immediate data.
12. (Original): The method of claim 1, further comprising:
 - packaging the data difference representations into a single data stream;
 - compressing the single data stream; and
 - storing the single data stream.
13. (Original): The method of claim 12, further comprising:
 - transmitting the single data stream;
 - uncompressing the single data stream; and
 - unpackaging the single data stream into the data difference representations.

14. (Previously presented): A system of reducing a size of data difference representations, the system comprising:

a network;

a first computer system coupled to the network;

a system memory coupled to the first computer system, wherein the system memory stores one or more computer programs executable by the first computer system;

wherein the computer programs are executable to:

identify an original version of an input data stream in an original form;

divide the original form of the original version of the input data stream into separate original version output data streams through the use of a pre-processor;

identify an updated version of the input data stream in the original form;

divide the original form of the updated version of the input data stream into separate updated version output data streams through the use of the pre-processor; and

difference at least two of the separate updated version output data streams with a corresponding separate original version output data stream to produce data difference representations.

15. (Original): The system of claim 14, wherein the data difference representations are smaller than a data difference representation created by differencing the original form of the updated version of the input data stream with the original form of the original input data stream.

16. (Original): The system of claim 14, further comprising:

a second computer system coupled to the network;

a system memory coupled to the second computer system, wherein the system memory stores one or more computer programs executable by the second computer system;

wherein the pre-processor is located in the first computer system; and

wherein the post-processor is located in the second computer system.

17. (Previously presented): The system of claim 16, wherein the computer programs are further executable to:

reconstruct the separate updated version output data streams from the data difference representations and the original version output data streams; and

combine the separate updated version output data streams into the original form of the updated version of the input data stream through the use of a post-processor.

18. (Original): The system of claim 16 wherein the original form of the original version of the input data stream is empty.

19. (Original): The system of claim 16 wherein the pre-processor comprises decompression algorithms.

20. (Previously presented): The system of claim 16 wherein the dividing separates volatile components of the input data stream from less volatile components of the input data stream.

21. (Currently amended): The system of claim 16 20 wherein the input data stream is executable code.

22. (Original): The system of claim 21 wherein the volatile components comprise branch targets.

23. (Original): The system of claim 21 wherein the volatile components comprise data addresses.

24. (Original): The system of claim 21 wherein the less volatile components comprise instruction code.

25. (Original): The system of claim 21 wherein the less volatile components comprise immediate data.

26. (Original): The system of claim 16, wherein the computer programs are further executable to:

package the data difference representations into a single data stream;
compress the single data stream; and
store the single data stream on a memory medium coupled to the first computer system.

27. (Currently amended): The system of claim 26, wherein the computer programs are further executable to:

~~package the data difference representations into a transmit the single data stream from the memory medium coupled to the first computer system to a second computer system over a computer system network;~~

~~compress uncompress~~ the single data stream; and

~~store unpackage~~ the single data stream into the data difference representations, ~~on a memory medium coupled to the first computer system.~~

28. (Previously presented): A carrier medium which stores program instructions, wherein the program instructions are executable to implement reducing a size of data difference representations comprising:

identifying an original version of an input data stream in an original form;

dividing the original form of the original version of the input data stream into separate original version output data streams through the use of a pre-processor;

identifying an updated version of the input data stream in the original form;

dividing the original form of the updated version of the input data stream into separate updated version output data streams through the use of the pre-processor; and

differencing at least two of the separate updated version output data streams with a corresponding separate original version output data stream to produce data difference representations.

29. (Original): The carrier medium of claim 28, wherein the data difference representations are smaller than a data difference representation created by differencing the original form of the updated version of the input data stream with the original form of the original input data stream.

30. (Previously presented): The carrier medium of claim 28, wherein the program instructions are further executable to implement:

reconstructing the separate updated version output data streams from the data difference representations and the original version output data streams; and

combining the separate updated version output data streams into the original form of the updated version of the input data stream through the use of a post-processor.

31. (Original): The carrier medium of claim 28 wherein the original form of the original version of the input data stream is empty.

32. (Original): The carrier medium of claim 28 wherein the pre-processor comprises decompression algorithms.

33. (Previously presented): The carrier medium of claim 28 wherein the dividing separates volatile components of the input data stream from less volatile components of the input data stream.

34. (Original): The carrier medium of claim 33 wherein the input data stream is executable code.

35. (Original): The carrier medium of claim 34 wherein the volatile components comprise branch targets.

36. (Original): The carrier medium of claim 34 wherein the volatile components comprise data addresses.

37. (Original): The carrier medium of claim 34 wherein the less volatile components comprise instruction code.

38. (Original): The carrier medium of claim 34 wherein the less volatile components comprise immediate data.

39. (Original): The carrier medium of claim 28, wherein the program instructions are further executable to implement:

- packaging the data difference representations into a single data stream;
- compressing the single data stream; and
- storing the single data stream on a memory medium coupled to a first computer system.

40. (Original): The carrier medium of claim 39, wherein the program instructions are further executable to implement:

- transmitting the single data stream from the memory medium coupled to the first computer system to a second computer system over a computer system network;
- uncompressing the single data stream; and
- unpackaging the single data stream into the data difference representations.

41. (Original): The carrier medium of claim 28, wherein the carrier medium is a memory medium.

42. (Previously Presented): The method of claim 1, wherein dividing the original form of the updated version of the input data stream into separate updated version output data streams includes parsing the input data stream according to a data type of the input data stream.

43. (Previously Presented): The system of claim 14, wherein dividing the original form of the updated version of the input data stream into separate updated version output data streams includes parsing the input data stream according to a data type of the input data stream.

44. (Previously Presented): The carrier medium of claim 28, wherein dividing the original form of the updated version of the input data stream into separate updated version output data streams includes parsing the input data stream according to a data type of the input data stream.

45. (Previously presented): The method of claim 1, wherein differencing at least two of the separate updated version output data streams with a corresponding separate original version

output data stream to produce data difference representations comprises differencing each of the separate updated version output data streams with a corresponding separate original version output data stream to produce data difference representations.

46. (New): The method of claim 6, wherein the volatile components are substantially dynamically changing components and wherein the less volatile components are substantially static components.

47. (New): The method of claim 1,

wherein the dividing separates first components of the input data stream from second components of the input data stream, and

wherein the first components are components of the input data stream that are statistically more likely to have changed between the original version of the input data stream and the updated version of the input data stream than the second components.

48. (New): The method of claim 47, wherein a pattern template is used to separate the first components and the second components.